

METHOD FOR RECONSTITUTING HARDWARE SET AND HARDWARE DEVICE TO BE RECONSTITUTED

Patent number: JP2000040745
Publication date: 2000-02-08
Inventor: ABRAMOVICI MIRON
Applicant: LUCENT TECHNOLOGIES INC
Classification:
- international: H01L21/82; G06F9/06
- european:
Application number: JP19990011395 19990120
Priority number(s): US19980010000 19980121

Also published as:



US6034538 (A1)
GB2333625 (A)

Abstract not available for JP2000040745

Abstract of corresponding document: **US6034538**

A set of reconfigurable hardware includes a number of field programmable gate arrays (FPGAs), a controller referred to as a page manager, and a RAM-based local memory. In an illustrative embodiment, each of the FPGAs is suitable for implementing any one of a number of different portions of a logic circuit. A netlist or other descriptive information characterizing the logic circuit is partitioned into a number of pages, each of the pages corresponding to one of the portions of the circuit. The page manager controls the loading and unloading of the pages from the local memory into the FPGAs of the reconfigurable hardware, and controls storage and transfer of inter-page signals. The page manager is configured to detect "page faults" such as, for example, an unloaded page with a full input buffer. The page manager responds to a given page fault by subsequently loading the previously unloaded page into one of the FPGAs. The page manager may include FIFO input buffers or other suitable sets of registers for storing inter-page signal values for loaded and unloaded pages.

~~~~~  
Data supplied from the **esp@cenet** database - Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号  
特開2000-40745  
(P2000-40745A)

(43) 公開日 平成12年2月8日 (2000.2.8)

|                           |       |               |           |
|---------------------------|-------|---------------|-----------|
| (51) Int.Cl. <sup>7</sup> | 識別記号  | F I           | テマコード(参考) |
| H 0 1 L 21/82             |       | H 0 1 L 21/82 | C         |
| G 0 6 F 9/06              | 5 4 0 | G 0 6 F 9/06  | 5 4 0 F   |

審査請求 未請求 請求項の数25 O L (全 10 頁)

(21) 出願番号 特願平11-11395

(22) 出願日 平成11年1月20日 (1999.1.20)

(31) 優先権主張番号 09/010000

(32) 優先日 平成10年1月21日 (1998.1.21)

(33) 優先権主張国 米国 (US)

(71) 出願人 596077259

ルーセント テクノロジーズ インコーポ  
レイテッド

Lucent Technologies  
Inc.

アメリカ合衆国 07974 ニュージャージ  
ー、マレーヒル、マウンテン アベニュー  
600-700

(74) 代理人 100081053

弁理士 三俣 弘文

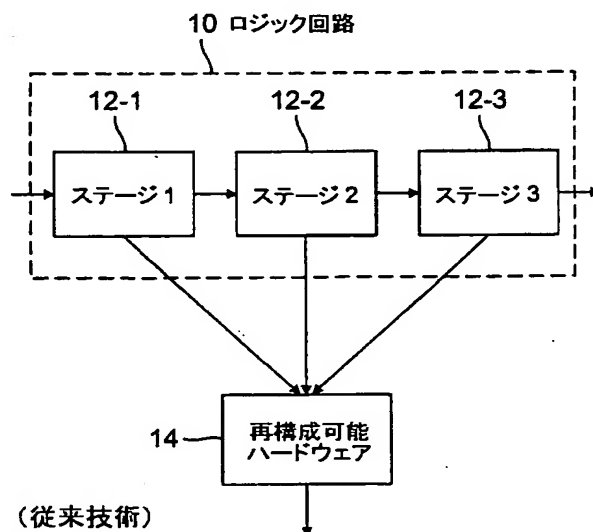
最終頁に続く

(54) 【発明の名称】 ハードウェアのセットを再構成する方法および再構成可能ハードウェア装置

(57) 【要約】

【課題】 適応化コンピューティングや他のアプリケーションにて用いる再構成可能ハードウェアにおいて、所望のいかなる大きさ、構造、複雑さを有する論理回路においても効率的な実装を可能にする再構成可能ハードウェア技術を提供する。

【解決手段】 再構成可能ハードウェアの容量を拡張する仮想論理システムを提供する。この仮想論理システムは部分的にハードウェアペーシングをベースとする。このハードウェアペーシングは、回路のアクティブ部分のみを再構成可能ハードウェアにロードされている状態に維持し、現在利用していない回路部分をメモリーに置き、これは必要があれば再構成可能ハードウェアへとダウンロードされる。本発明の仮想論理システムは、回路の大きさや同期回路構造の種類を問わず、また非常にユーザ透過的な方式で動作する。



## 【特許請求の範囲】

【請求項 1】 回路の異なる部分を実装するためにハードウェアのセットを再構成する方法であって、

(A) 複数のページへと回路のモデルを区分するステップと、

前記ページそれぞれは、回路の部分のうちの 1 つに対応しており、

(B) メモリからハードウェアへのページのシーケンシャルロードを制御するステップとを有し、

アンロードされたページに関する状態の検出により、その後のそのアンロードされたページのロードをトリガーすることを特徴する方法。

【請求項 2】 アンロードされたページに関する前記状態は、そのアンロードされたページに対応する入力バッファが実質的に満杯であることの指示に対応することを特徴する請求項 1 記載の方法。

【請求項 3】 前記制御するステップ (B) は、

(C) 当該ページに対応する FIFO バッファへとページ間信号値を記憶するステップを有することを特徴する請求項 1 記載の方法。

【請求項 4】 前記再構成可能ハードウェアは、複数の FPGA を具備し、前記制御するステップ (B) は、

(D) 前記各 FPGA へと各ページのサブセットがロードされるように、各ページのロードを制御するステップを有することを特徴する請求項 1 記載の方法。

【請求項 5】 前記制御するステップ (B) は、

(E) 当該ページが異なる FPGA へと異なる時間にてロードされるように、各ページのロードを制御するステップを有することを特徴する請求項 4 記載の方法。

【請求項 6】 前記制御するステップ (B) は、

(F) 再構成可能ハードウェアの第 1 部分へとロードされた第 1 ページと第 2 ページに対応する入力バッファとの間の接続を確立するステップを有することを特徴する請求項 1 記載の方法。

【請求項 7】 前記制御するステップ (B) は、

(G) 置換元ページに対応する再構成可能ハードウェアの部分における入力および出力線の切断を指示するステップと、

(H) 置換元ページとは別のページに対応する、前に記憶したページ間信号値を取り出すステップと、

(I) 前記別のページに対応する入力および出力線の接続を指示するステップとを有することを特徴する請求項 1 記載の方法。

【請求項 8】 (A) それぞれが回路の異なる部分を独立に実装することができる複数の再構成可能デバイスと、

前記回路は、それぞれが前記回路の各部分と対応する複数のページへと区分されたモデルにより特徴づけられ、

(B) 前記再構成可能デバイスとつながれ、各ページのロードを制御するように動作するコントローラとを有

し、

これにより、アンロードされたページに関する状態の検出により、その後のそのアンロードされたページのロードをトリガーさせることを特徴する再構成可能ハードウェア装置。

【請求項 9】 アンロードされたページに関する前記状態は、そのアンロードされたページに対応する入力バッファが実質的に満杯であることの指標に対応することを特徴とする請求項 8 記載の再構成可能ハードウェア装置。

【請求項 10】 (C) 前記再構成可能デバイス (A) および前記コントローラ (C) につながれたローカルメモリを有することを特徴とする請求項 8 記載の再構成可能ハードウェア装置。

【請求項 11】 前記コントローラ (C) は、ページ間信号値をそのページに対応する FIFO バッファ内へと記憶するように動作することを特徴とする請求項 8 記載の再構成可能ハードウェア装置。

【請求項 12】 前記再構成可能デバイス (A) は、フィールドプログラマブルゲートアレイであり、前記コントローラ (C) は、各ページのサブセットが前記 FPGA へとロードされるように、各ページのロードを制御するように動作することを特徴とする請求項 8 記載の再構成可能ハードウェア装置。

【請求項 13】 前記コントローラ (C) は、各ページが異なる時間において異なる FPGA へとロードされるように、各ページのロードを制御するように動作するを有することを特徴とする請求項 12 記載の再構成可能ハードウェア装置。

【請求項 14】 前記コントローラ (C) は、前記再構成可能デバイス (A) の第 1 デバイスへとロードされた第 1 ページと第 2 ページに対応する入力バッファとの間の接続を確立するように動作することを特徴とする請求項 8 記載の再構成可能ハードウェア装置。

【請求項 15】 前記コントローラ (C) および少なくとも前記再構成可能デバイス (A) のサブセットは、ホストコンピュータへバスを介してつながれた回路カード上に実装されることを特徴とする請求項 8 記載の再構成可能ハードウェア装置。

【請求項 16】 前記コントローラ (C) は、パーソナルコンピュータの資源を少なくとも一部用いて実装されるを有することを特徴とする請求項 8 記載の再構成可能ハードウェア装置。

【請求項 17】 回路の異なる部分を実装するためにハードウェアのセットを再構成する装置であって、

(A) 複数のページへと回路のモデルを区分する手段と、各ページは、当該回路の部分のうちの 1 つに対応し、

(B) メモリからハードウェアへのページのシーケンシャルロードを制御する手段とを有し、

これにより、ある時間において複数のページがハードウェア内に実装されることを特徴とする装置。

【請求項 18】 回路の異なる部分を実装するためにハードウェアのセットを再構成する装置であって、

(A) 複数のページへと回路のモデルを区分する手段と、

各ページは、当該回路の部分のうちの 1 つに対応し、

(B) メモリからハードウェアへのページのシーケンシャルロードを制御する手段とを有し、

これにより、アンロードされたページに関する状態の検出によりその後のそのアンロードされたページのロードをトリガーすることを特徴とする装置。

【請求項 19】 (A) 再構成可能ハードウェアを用いて回路の異なる部分を実装する手段と、

前記回路は、複数のページにより特徴づけられ、前記ページそれぞれは、前記回路の部分のうちの 1 つに対応し、

(B) 前記再構成可能ハードウェアへと各ページのシーケンシャルロードを制御する手段とを有し、

これにより、前記ページの複数のページが、ある時間において前記再構成可能ハードウェアにおいて実装されることを特徴とする再構成可能ハードウェア装置。

【請求項 20】 (A) 再構成可能ハードウェアを用いて回路の異なる部分を実装する手段と、

前記回路は、複数のページにより特徴づけられ、前記ページそれぞれは、前記回路の部分のうちの 1 つに対応し、

(B) 前記再構成可能ハードウェアへと各ページのシーケンシャルロードを制御する手段とを有し、

これにより、アンロードされたページに関する状態の検出によりその後のそのアンロードされたページのロードをトリガーすることを特徴とする再構成可能ハードウェア装置。

【請求項 21】 回路の異なる部分を実装するためにハードウェアのセットを再構成する方法であって、

(A) 回路の異なる部分を実装するために複数の再構成可能デバイスを用いるステップと、

前記回路は、複数のページにより特徴づけられ、前記ページそれぞれは、前記回路の部分の 1 つに対応し、

(B) 前記再構成可能デバイスへと各ページのシーケンシャルロードを制御するステップとを有し、

これにより、前記ページの複数のページが、ある時間において前記再構成可能ハードウェアにおいて実装されることを特徴とする方法。

【請求項 22】 回路の異なる部分を実装するためにハードウェアのセットを再構成する方法であって、

(A) 回路の異なる部分を実装するために複数の再構成可能デバイスを用いるステップと、

前記回路は、複数のページにより特徴づけられ、前記ページそれぞれは、前記回路の部分の 1 つに対応し、

(B) 前記再構成可能デバイスへと各ページのシーケンシャルロードを制御するステップとを有し、

これにより、ロードされたページからアンロードされたページへ向かうページ間信号が、その後のそのアンロードされたページのロードまでに少なくとも一時的に記憶されることを特徴とする方法。

【請求項 23】 再構成可能ハードウェアにおいて回路を実装する装置であって、

(A) 前記再構成可能ハードウェアにより実装することができるパイプライン化できない各ページへと回路を区分する回路パーティショナーと、

(B) 前記パイプライン化できない各ページのうちの現在実装されているページとそれらページのうちの別のページとの間で結果を交信するためのメモリとを有することを特徴とする装置。

【請求項 24】 再構成可能ハードウェアにおいて回路を実装する装置であって、前記回路は、各ページへと区分され、各ページのうちの最大大きさのページを前記再構成可能ハードウェア上に実装することができ、

(A) 前記再構成可能ハードウェアを各ページへと任意の順序で制御状態で供給するページマネージャと、前記ページマネージャは、前記回路を実装するために各ページをスワッピングし、

(B) 各ページのうちの現在実装されているページとそれらページのうちの別のページとの間で結果を交信するためのメモリとを有することを特徴とする装置。

【請求項 25】 (A) それぞれが回路の異なる部分を独立に実装することができる複数の再構成可能デバイスと、

前記回路は、それぞれが前記回路の各部分と対応する複数のページへと区分されたモデルにより特徴づけられ、

(B) 前記再構成可能デバイスとつながれ、各ページのロードを制御するように動作するコントローラとを有し、

これにより、前記ページの複数のページが、ある時間において再構成可能ハードウェアにおいて実装されることを特徴する再構成可能ハードウェア装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、適応化コンピューティングや他のアプリケーションにて用いる再構成可能ハードウェアに関し、特に、所望のいかなる大きさ、構造、複雑さを有する論理回路においても効率的な実装を可能にする再構成可能ハードウェア技術に関する。

【0002】

【従来の技術】再構成可能ハードウェアは論理回路機能を実装するために適応化コンピューティングや他のアプリケーションにおいて用いられている。フィールドプログラマブルゲートアレー (FPGA: field programmable gate array、現場でプログラム可能なゲートアレー)

や他の同様なデバイスをベースにしてもよい再構成可能ハードウェアは、別の時に別の論理機能を提供するように再構成することができ、通常は相当に多くのハードウェアを必要とってしまう複合回路の機能を提供することができる。FPGAをベースにする再構成可能ハードウェアは、文献、J.H. Mayer, "Reconfigurable computing redefines design flexibility", Computer Design, pp. 49-52, February 1997, J. Rosenberg, "Implementing Cache Logic(TM) with FPGAs", Atmel Application Note 0461A, pp. 7-11~7-14, B.K. Fawcett, "Applications of Reconfigurable Logic," in "more FPGAs", W. R. Moore, W. Luk, eds., Abingdon EE & CS Books, Oxford, England, pp. 57-69, 1994, などに説明されている。

#### 【0003】

【発明が解決しようとする課題】典型的な適応化コンピューティングのアプリケーションにおいて、回路モデル(ネットリスト(netlist)と呼ばれる)を作るのに用いるためにプリプロセッシング段階が用いられる。回路モデルは、再構成可能ハードウェアへとロードされ、それが特定の回路の機能を提供するようにされる。しかし、従来の再構成可能ハードウェアのプラットフォームは、通常の複雑さの回路に対するネットリスト全体を扱う容量が不十分なことが多い。エミュレータと呼ばれる種類の再構成ハードウェアは非常に大きな容量を有し、約600万ゲートを収容するが、エミュレータは非常に高価であり、通常ゲートあたり100円程度もするため、多くの重要なアプリケーションには使えない。

【0004】このような高価なエミュレータの利用をさけるため、回路を複数のステージへと分け、ステージそれぞれにはそれほど高価ではない再構成可能ハードウェアプラットフォームを用いるアプリケーションがあり、図1にその例を示す。ロジック回路10は、3つの別々のステージ12-1、12-2、12-3を有する。ロジック回路10のこれらステージ12-1、12-2、12-3それぞれは、同じ再構成可能ハードウェア14を用いて実装される。これらステージのうちの1つのステージの特徴を表す情報を含む部分的ネットリストが再構成可能ハードウェア14へとロードされ、そのステージに対して再構成可能ハードウェアが処理動作を行えるようにする。この方法により、再構成可能ハードウェア14はステージ12-1~12-3により共有され、ステージそれぞれに必要な特定の処理動作を行うために周期的に再構成される。図1の技術は複数のステージへと自然に区分することができるようなアプリケーションにおいて適している。このアプリケーションでは、各ステージはパイプラインのステージと同様な方法により、前のステージにより準備されたデータを用いる。例えば、文献、H. Schmit, "Incremental Reconfiguration for Pipelined Applications", IEEE Symposium on FPGAs for

Custom Computing Machines, pp. 47-55, 1996, などを参照されたい。

【0005】図1に示したようなパイプライン化された回路構造のステージそれぞれにて用いる論理機能を想定すると、このようなアプリケーションは各ステージに対してハードウェアを再構成することによりハードウェアを再利用できる。しかし、この種のアプローチには多くの問題がある。例として、論理回路を複数のステージへと区分することには、大量の高コストな人間の設計労力が必要となってしまう。また、多くの重量なアプリケーションは、識別可能なステージへと区分することができるパイプライン化構造を示さない。回路がパイプライン化構造を示したとしても、この構造は大きなネットリストから認識してもらうのが困難であったり、1つのステージを利用可能なハードウェアが収容できるような大きさの回路ではなかったりする。このように、現存する再構成可能ハードウェアは、経済的な方法により大きな回路のネットリストを処理する能力がなく、多くの重要なアプリケーションにおいて用いることができない。

#### 【0006】

【課題を解決するための手段】本発明は、再構成可能ハードウェアの容量を拡張する仮想論理システムを提供する。この仮想論理システムは部分的にハードウェアページングをベースとする。このハードウェアページングは、回路のアクティブ部分のみを再構成可能ハードウェアにロードされている状態に維持し、現在利用していない回路部分をメモリーに置き、これは必要があれば再構成可能ハードウェアへとダウンロードされる。上述の従来技術とは対照的に、本発明の仮想論理システムは、回路の大きさや同期回路構造の種類を問わず、また非常にユーザ透過的(user-transparent)な方式で動作する。

【0007】図示した実施例において、再構成可能ハードウェアのセットは、多くの現場でプログラム可能なゲートアレイ(FPGA)、ページマネージャ(PAGMAN)と呼ぶコントローラ22、FPGAやPAGMANにつながったRAMベースのローカルメモリー24を具備する。各FPGAは論理回路の異なる部分を実装するのに適している。論理回路を特徴づける初期ネットリストは多くのページへと区分され、各ページは回路の部分のうちの1つに対応する。各FPGAは、これらページのうちの1つの回路機能を実装する。PAGMANは、これらページの再構成可能ハードウェアのFPGAへのロードとアンロードを制御し、ロードされたページとアンロードされたページとの間のページ間信号の記憶および転送を制御する。PAGMANは、FIFO入力バッファや他の適切なレジスタセットを具備することができ、ページ間信号値を記憶し、ページ間通信を制御する。PAGMANは、1もしくは複数個の異なる種類の「ページフォールト(ページングの失敗)」を検出するように構成する。このページフォールトは、例として、

アンロードされたページの入力バッファが満杯（フル）となった場合に起きてしまう。PAGMANは、ページフォールトに対して、その後にFPGAのうちの1つへとアンロードされたページをロードすることにより応答させることができる。FPGA、PAGMAN、ローカルメモリは、ホストコンピュータへとバスを介してつながった回路カード上で実装できる。

【0008】ランタイム（実行時）再構成を活用することにより、仮想論理システムは再構成可能ハードウェアの物理的容量よりもはるかに大きな回路の実装を可能にし、かつ、ベースとするハードウェアや実行時インフラ（基盤技術）からユーザを隔離し解放することができる。本発明は、回路の大きさ、構造、複雑さに関わらず、広範囲の重要なアプリケーションにおいて再構成可能ハードウェアを利用することを経済的に実現可能とした。

#### 【0009】

【発明の実施の形態】以下の説明において、本発明は、複数のFPGAを具備し、非常に大きなネットリストを有する複雑な論理回路を処理するのに適した再構成可能ハードウェアのセットを用いるが、本発明は再構成可能ハードウェアの種類によって限定されない。例として、1つのFPGAをベースとする再構成可能ハードウェアのアプリケーションに容易に適合させることができるが、FPGAを具備しない再構成可能ハードウェアを用いてもよい。本明細書において用語「ページ」は、再構成可能ハードウェアにおいて実装される回路の少なくとも部分を特徴づけるデータその他の情報のいずれをも含むように意図している。用語「コントローラ」は、ページング機能や他の仮想論理機構の少なくとも一部を行うように構成されるようなページマネージャや他の処理デバイスをいうように意図している。用語「回路モデル」は、回路を特徴づけるネットリストや他の情報の種類を含むように意図している。

【0010】本発明は、相当にユーザ透過的な方法により、所望のいかなる大きさ、いかなる同期回路構造の種類である論理回路を実装することを再構成可能ハードウェアのセットに対して可能にさせる仮想論理システムを提供する。本明細書において、論理回路は初期ネットリストと呼ぶモデルにより特徴づけられ、これは任意の大きさであってよい。このネットリストは、論理合成や回路図把握(schematic capture)のような周知の異なる種類のいかなる数の設計方法によって生成してもよい。論理回路のネットリストは、幾つかのばらばらなサブ回路へと区分され、これらサブ回路それぞれは「ページ」と呼ぶ初期ネットリストの部分により表される。このようなサブ回路への論理回路の区分化は、初期ネットリストを対応するページへと区分することにより行われる。上述のような従来技術のパイプライン化構造の複数のページへの区分化とは対照的に、本発明に従う初期ネット

リストのページへの区分化は異なるページ間でいかなる特定の機能的関係をも必要としない。回路に適用される再構成可能ハードウェアの効率を改善するため、再構成可能ハードウェアにおいて演算と再構成周期との釣り合いをとるためにページ区分化アルゴリズムを用いることができる。

【0011】サブ回路は、その対応するページをハードウェアへとロードすることにより再構成可能ハードウェアの一部として実装させることができる。再構成可能ハードウェアにサブ回路が実装された「アクティブ」ページは、ロードされたページと呼ぶ。現在利用されていないページはアクティブではなく、アンロードされたページと呼ぶ。アクティブでないページは再構成可能ハードウェアに関連づけられたローカルメモリ内に必要となるまで記憶される。図2を用いて下で詳細に説明するページマネージャ（PAGMAN）は、再構成可能ハードウェアに出入りするようなページのロードとアンロードを制御するのに用いられ、アクティブでないページをアンロード状態でローカルメモリ内で記憶されている状態に保ち、ページフォールトを検出し処理する。

【0012】図2には、再構成可能ハードウェア20のセットを示し、これは、上述のような方法でページのロードとアンロードを制御することができるページマネージャ22を具備する。再構成可能ハードウェア20はFPGA1、FPGA2、FPGA3、FPGA4の4つのFPGAを有する。これら4つのFPGAのそれぞれが論理回路ネットリストのページ1つと対応しているものと想定することができる。4つのFPGAそれぞれはバス23を介してここではRAMであるローカルメモリ24と通信する。ページマネージャ22は、各FPGAを制御し、またローカルメモリ24と通信するようにバス23へとつながっている。ローカルメモリ24はページそれぞれに対して構成情報やレジスタの値を記憶させるための専用領域を具備していてもよい。この実施例におけるローカルメモリ24は4つのFPGAそれぞれにより共有され、異なるFPGAにおいて異なる時に同じページがロードできるようにされている。図2の実施例において、本発明にとっては必要条件ではないが、FPGA資源に関してページそれぞれは大まかに同一の大きさであることを想定することができる。動作時の異なる時において、元々の回路ネットリストの異なるページを同じFPGAが持っていることができる。

【0013】演算処理が現在にロードされているページのみに対して進行するので、ページごとの「ローカルタイム」のようなタイミング情報をページマネージャ22が追跡することができる。同期サブ回路に対応するページのローカルタイムは、再構成可能ハードウェアにおいて当該ページの動作開始から経過したクロックサイクルの数を一般には表す。結局全てのページのローカルタイムはほぼ同じとなるが、しかし、初期動作時では、これ

らのローカル時間は、「同期ずれ状態(out-of-sync)」であって、例えば、ロードされたページに対してのみローカルタイムが進行するようになることがあり得る。適切な動作を確実にするため、時間  $t$  の間は、その時間  $t$  の間に対応するその入力データの全てが利用可能となるまでは、演算を開始することを許されるページがあつてはならない。

【0014】図2のマルチFPGA再構成可能ハードウェア20は、バス(PCI(peripheral component interconnect)バスなど)を介してホストコンピュータへと取り付けられた単一のコプロセッサ回路ボード上で実装させることができる。FPGAは、Atmel Corporation(米国、San Jose, CA)から得られ、上記文献にて説明されているような市場で得られるFPGAであつてもよい。他の適切なFPGAとして、Altera(San Jose, CA)、Lattice Semiconductor(Hillsboro, OR)、Lucent Technologies(Murray Hill, NJ)、Motorola(Chandler, AZ)、Xilinx(San Jose, CA)のような米国のメーカーから得られるものがある。また、多くの代替構成を用いることもできる。例として、再構成可能ハードウェア20は単一のFPGAや、FPGAではない多くの種類のハードウェアで構成してもよい。また、再構成可能ハードウェアはホストコンピュータに取り付けられた回路ボードではなく、スタンドアロン型コンピュータや他のプロセッシングデバイスとして実装することができる。

【0015】再構成可能ハードウェア20内のローカルメモリ24は、ホストコンピュータ内のRAM、再構成可能ハードウェア内の専用RAM、あるいはこれらの組み合わせであつてもよい。ここで、図2の実施例におけるローカルメモリ24の大きさは記憶できるページ数、そして処理できる論理回路の大きさを制約することとなる。再構成可能ハードウェア20内で処理できるページ数を増やすためにディスクベースの記憶手段などの他の種類の記憶手段を用いてもよいが、このようなデバイスからのページの記憶および取り出しによってハードウェア動作を遅くしてしまうことがある。従つて、RAMベースのローカルメモリが処理速度が重要なアプリケーションにおいては好ましい。

【0016】ページマネージャ22は、再構成可能ハードウェア20内の4つのFPGAそれぞれのページのロードとアンロードを制御する。ある時間においてどのページがどのFPGAへとロードされてるかの情報を記憶し、またページのマッピングやそれらのローカルメモリ24内の位置を記憶する。またページマネージャ22は、ロードされたページの間の信号をルーティングし、ページフォールトを検出する。ページマネージャ22は1もしくは複数のFIFOバッファを含み、ページ間信号値の記憶及び転送を制御するのに用いる。FIFOバッファを用いると、ページがアクティブになる前に多くの信号値をページマネージャ22が蓄積することを可

能にし、ページ構成をダウンロードしたときに有用な機能をより迅速に行うことができるようになる。このように実装することにより、FPGAへとページ構成をダウンロードするのに費やす時間とページ構成をダウンロードした後のFPGAにおける演算時間との間の不釣り合いを減少することができる傾向にある。

【0017】図3は、FIFO入力バッファ30の一部を示してあり、これはページAに対するページマネージャ22内に具備することができる。ページマネージャ22は再構成と演算時間との釣り合いをとるように試みる。これは、ロードされたページが別のページのために空きを作るようにアンロードされる前に、なるべく多くの演算を行うべきであることを意味する。このため、ページAの入力データベクトルは図3に示すようにバッファリングされる。ページB、C、Dが生成した、ページAに対するデータベクトルの異なる「スライス」ないし部分は、パラレル(並列)FIFOバッファに同期されて蓄積される。図3のFIFO入力バッファ30は、3つのセットのパラレルFIFOバッファ32、34、36を有し、それぞれページB、C、Dにより生成されたデータベクトルを保持する。FIFOバッファ32、34、36のセットそれぞれは、いくつかの個々のスライスバッファ33、35、37を具備する。ページCによるデータベクトルの計算は、そのバッファセットの相対的占有度により示されるように、ページBおよびDによる計算よりも一時的に遅れる。理想的にはAはその入力バッファ30が満杯になった後にのみロードされるのがよい。これは、一般化したデータフローモデルの一種として考えることができる。このモデルでは、ページAに必要とされる演算数(オペランド)はページB、C、Dが生成したベクトルスライスに対応し、最後の $k$ のクロックサイクルで演算されたオペランドを演算は待つ。ここで、 $k$ は、セット32、34、36それぞれにおけるバッファの数である。ページマネージャ22は、回路のページのそれぞれに対して図3で示したものと同様な入力バッファを維持する。なお、本発明は、他の種類のページ間バッファリングをも用いることができる。

【0018】図4は、図2のページマネージャ22にて実装することができるページフォールト処理動作の流れ図である。アンロードされたページの入力バッファが満杯になったことを表すページフォールトを検知すると、ページマネージャ22は、そのページフォールトを直すために置換元ページ(置換すべきページ)を選択する(50)。選択されるページは、例えば、最近最もロードされていないページ、あるいは他の所定規則を用いて選択される。代わりに、ページ置換アルゴリズムを特定の基準を最小化するように置換ページを選択して用いてもよい。次に、ページマネージャ22は置換元ページの全てのレジスタの値をローカルメモリ24内に保存する(52)。ページマネージャ22は置換元ページのFPGA

## 11

における全ての入出力（I/O）ピンを切断する（54）。次に、新しく置換するページ（新置換ページ）の構成情報をページマネージャ22がローカルメモリ24から置換元ページが占有するFPGAへとダウンロードする（56）。新置換ページのレジスタの値はローカルメモリ24から取り出され、復帰される（58）。新置換ページのFPGAのI/Oピンを接続する（60）。これにより、新置換ページの入力バッファから対応するFPGAの入力へ、またそのFPGAの出力から新置換ページが供給するページの入力バッファへの必要な接続を確立する。

【0019】上述の方法によりロードされたページは、そのベクトルのスライス全てがロードされたページの入力バッファに到達してからすぐにその次の入力ベクトルを処理する。ページマネージャ22は、そのページ入力バッファの状態を用いて次のページを取り替えて追い出し（スワップアウト）、演算のフローを進行させる。例えば、すでに満杯な入力バッファの少なくとも1つを供給（フィード）するページは、アンロードされる。なぜなら、その次の出力ベクトルのための記憶場所がないからである。完全に満杯なページ入力バッファがなければ、スワップインすべき次のページは、処理すべきベクトルの数が最も多い入力バッファを有するページとして選択される。

【0020】ページマネージャ22は、マイクロプロセッサ、FPGA、これらの組み合わせ、FPGAと現場でプログラム可能な相互接続チップ（FPIC）との組み合わせを用いて実装することができる。また他のハードウェア、ソフトウェア、ファームウェア、または図2〜4と関連して上述したページ管理機能を提供するのに適切なこれらの組み合わせを用いることができる。

【0021】ネットリストあるいは他の回路表現を各ページへと区分することを以下に説明する。図5Aは、フィードフォワード構造を具備する回路のページ構成を示すブロック図である。従って、図5Aのフィードフォワード構造における4つのページA、B、C、Dの間の相互接続は、ページの内部構造は1もしくは複数のフィードバックループを具備するかもしれないが、フィードバックループを全く具備しない。上述のバイブライニング構造は、フィードフォワード構造の一種である。プライマリ入力（PI）のみにより供給されるページは、独立ページと呼ぶ。例えば、図6Aにおいて、ページAおよびBは独立ページであり、ページCおよびDは従属ページである。独立ページは、任意の大きな数の入力データベクトルをバッファリングすることができる。

【0022】「ページレベル」と呼ばれる概念は以下のように循環的に定められる。PIのみにより供給されるページはレベル0を有するとされる。他の全てのページXのレベルはXに供給するページの中の最大レベルよりも大きなレベルである。図5において、ページAおよび

## 12

Bのレベルは0であり、ページCのレベルは $0+1=1$ であり、ページDのレベルは $1+1=2$ である。このページレベル情報をページマネージャ22が用いて、利用可能なFPGAにてロードするために各ページをスケジューリングする。例えば、一実施例において、同時に2ページのみをロードすることができるのとすると、ページマネージャ22はまずレベル0のページ（ページAおよびB）をロードし、PIからバッファリングされた入力データベクトルでこれら2ページの演算を開始する。次に、ページAおよびBは、それらの出力データベクトルをページCおよびDの入力バッファにて記憶する。

【0023】ページCの入力バッファが満杯になると、ページマネージャ22はページフォールトを認識する。次に、レベル0のページの1つ、ページAなどがそのFPGAからアンロードされ、（レベル1の）ページCそのFPGAへとロードされる。次に、ページCは、その入力バッファからのデータの処理を開始し、ページDの入力バッファにその結果を記憶する。この動作に伴い、（レベル2の）ページDに必要な構成およびレジスタの値がページBにより以前に用いられていたFPGAへとロードされる。ページDがロードされると、その入力バッファからのデータ処理を開始する。ページCがその入力バッファをからにするとすぐ、ページCのFPGAはページAへと再割り当てされる。ここで、ページAのロード処理はページDの動作と協調して動作する。ページAはそのロード処理を完成するとすぐに入力データベクトルの処理を開始することができる。

【0024】本発明に従うページパーディショナーは、ページ間論理動作を最小化するように試み、各ページは可能な限り「自らに含まれる」すなわち自律性（オートノマス）を有するようにされる。ページがより自律性を有すると、他のページが作ったデータをより多く必要とする前までは、その入力バッファに蓄積されたデータに関して行うことができる演算の量は増える。従って、パーディショナーの1つの目的として、各ページの内部にフィードバックループを保持することがある。FPGAの大きさに依存して、この目的は全ての回路において達成できるとは限らない。

【0025】図5Bは、図5Aのページ相互接続とともに、ページDの出力からページCの入力へのフィードバック（帰還）ループを示している。フィードバックループのためにページレベルは上述の手順を用いてはページCおよびDに対して演算することができない。従って、「規定されていない」ページレベルを用いて、フィードバックループを識別するのに用いることができる。FPGAの大きさの制約のため、ページCおよびDは同じページ内にともにグルーピングすることはできないと考えることができる。図5Bのページ間帰還ループを取り扱う方法として、2つの隣接するFPGAへと収まる必要があるように、1つの超ページ（スーパーページ）C—



13

DとしてページCおよびDを扱う方法がある。もちろん、グローバル帰還ループの存在は、回路全体が超ページとして扱われることを必要とし、これは、利用可能な再構成可能ハードウェアへと収まるためには大きすぎることになる。このような回路の問題としては、新しい再構成を必要とする前に、わずかなベクトルのみをバッファリングすることができるという点がある。この問題は、例えば、それらを必要とする前に構成のダウンロードをするための拡張ルックアップヘッドの技術、また、再構成可能ハードウェア内のページをエミュレーションすることなしに学習したページの動作の再生を可能とするキャッシング技術を用いることにより解決することができる。

【0026】本発明は、いかなる同期回路構造に適用することができ、ユーザにとって透過性のある方法で動作する再構成可能ハードウェア技術を提供する。従って、本発明は、上述の従来技術の問題となっている「パイプラインのみ」による構造的制約を有効に除去することができる。また、ユーザにとって、再構成可能ハードウェア上を回路がどのように実行、エミュレーション、あるいはシミュレーションするかを知らなくてもすみ、また、回路の大きさと利用可能なハードウェアの容量とのミスマッチの可能性について気にしないでもすみ。すなわち、本発明に従う仮想論理システムは、用いるハードウェアや実行時インフラからユーザを隔離し、ユーザがその設計を動作させる論理設計ではなく、動作せたいアプリケーションに焦点を当てることができる。

【0027】上述の実施例は、例示的な目的で示しており、本発明は他の構成も可能である。別の種類の再構成

14

可能ハードウェアやページング制御機構を用いる構成も可能である。また本発明は所望な回路のいかなる大きさをも収容するために自由にスケーリングすることができ、例えば、ローカルメモリ24の大きさ、FPGAの数や大きさを増やすことができる。また、回路に対してエミュレーションされたメモリとして用いるために埋め込みメモリを具備するようにFPGAを構成することができる。さらに、インクリメント型実行時再構成機能をサポートするように構成してもよい。

#### 10 【図面の簡単な説明】

【図1】 マルチステージパイプラインか回路構造で用いるのに適した従来技術の再構成可能ハードウェア技術を示すブロック図。

【図2】 本発明に従った仮想論理機構を具備する再構成可能ハードウェアのセットのブロック図。

【図3】 図2の再構成可能ハードウェアにて用いるのに適したページマネージャ（PAGMAN）におけるFIFOバッファの動作を示す説明図。

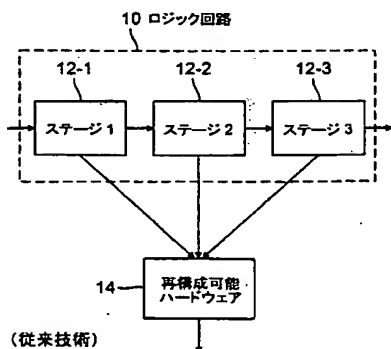
【図4】 本発明に従うPAGMANにて実装することができるページフォールト処理動作を示す流れ図。

【図5】 本発明のページ区分の原理を示すページブロック図。

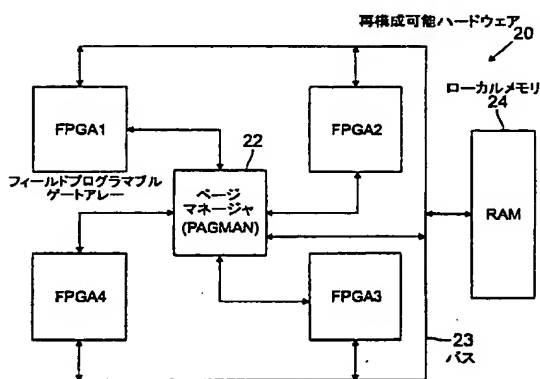
#### 【符号の説明】

- 10 ロジック回路
- 14、20 再構成可能ハードウェア
- 22 ページマネージャ（PAGMAN）
- 24 ローカルメモリ
- 30 FIFO入力バッファ
- 33、35、37 スライスバッファ

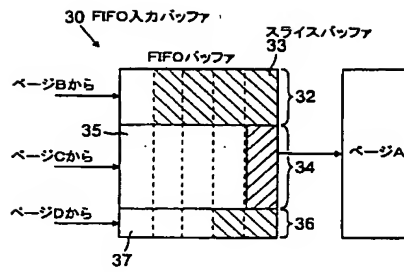
【図1】



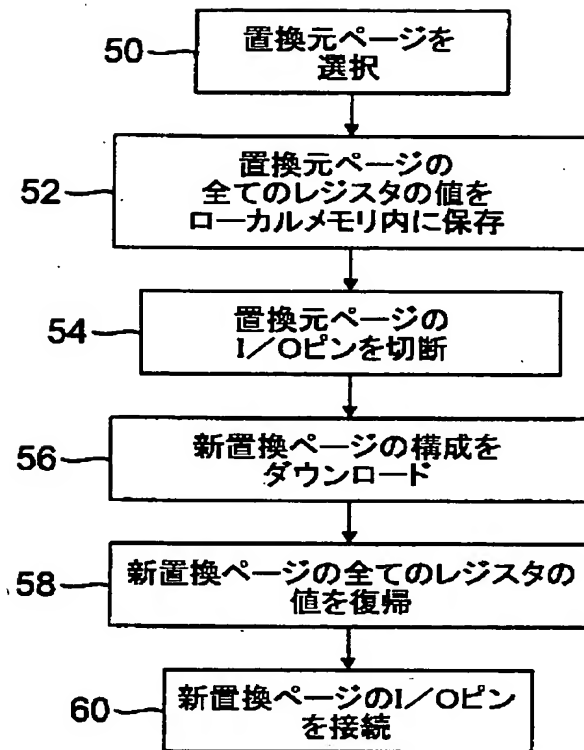
【図2】



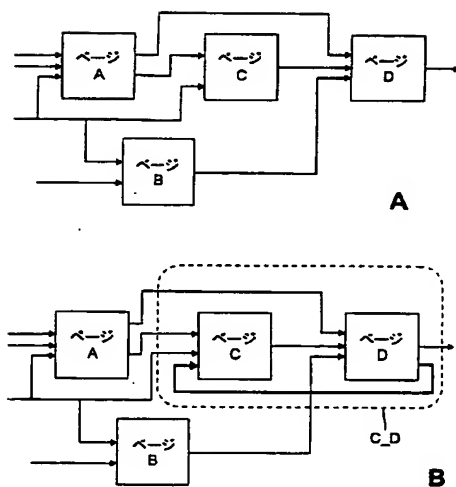
【図3】



【図4】



【図5】



フロントページの続き

(71)出願人 596077259  
600 Mountain Avenue,  
Murray Hill, New Je  
rsey 07974-0636 U. S. A.

(72)発明者 ミロン アブラモヴィチ  
アメリカ合衆国, 07922 ニュージャージ  
ー, バークレイ ハイツ, ノース ロード  
135